A study on Quantum Neural Network

Van Tien Nguyen
tiennvuit@gmail.com
February 9, 2026

***Abstract*—In this project, we delved into the quantum neural network (QNN) model, an emerging field in recent years. The appeal of this subject lies at the intersection of machine learning on quantum hardware. We introduced the theory and components of the Variational Quantum Algorithm, which serves as the foundation for quantum neural networks. Subsequently, we conducted experiments using traditional machine learning models, including Naive Bayes, SVMs, DNNs, and VQCs (a type of QNNs), across various benchmarks. Numerical results revealed that the practical performance of VQCs still lags behind that of traditional machine learning models and deep learning models. However, it is a promising approximation, indicating the need for further investigation to enhance VQC performance on real-world datasets.**

I. Introduction

Quantum Machine Learning [1] [2] has been a rapidly developing research field in recent years. Thanks to the advantages of quantum computing which is well-known as speeding up exponentially compared with classical algorithms, it would be a promising field in leading AI research. Among candidates, Quantum Neural Networks (QNNs) [3] are a type of quantum machine learning model that leverages the principles of quantum mechanics to process information. They are designed to harness the unique properties of quantum systems, such as superposition and entanglement, to perform computations at a scale that is currently unattainable with classical neural networks. QNNs have shown great potential in various fields such as quantum neural computing [3], generative learning [4], logistic applications [5].

Variational Quantum Circuit [6] (VQC) is one of the type of QNNs that is known through existing works that have theoretically shown the effectiveness and efficiency of VQC in the perfect scenario [7]. This project explore the following aspects on quantum neural network including theoretical and practical implementation:

- Study VQC framework and then implement it using Qiskit IBM library[1]
- Evaluate the performance of the VQC on various benchmark datasets including MNIST, Fashion-MNIST, and synthesis datasets.

Thereafter, as an future work, the CompVQC framework [8] was given in appendix C-A, which showing a way to compress QNN into smaller for more efficiency.

II. Backgrounds

In this section, the introduction for quantum computing, machine learning, and more important one, Quantum Machine Learning would be introduced. The main difference between classical computing and quantum computing is the computational unit, which is the quantum bit (qubit) in quantum devices, and bit in classical computers. This leads to the different characteristics in computation when people doing research on applying quantum devices to solve classical problems. Also, as a requirement, the quantum machine learning contains several additional steps compare with classical counterparts: quantum encoding part which maps the classical information to quantum devices, and quantum decoding part to classical information.

A. Basics in Quantum computing

In quantum computing [9], the basic unit of quantum information is a quantum bit or qubit. A single-qubit pure state is described by a unit vector in the Hilbert space $\mathbb{C}^2$, which is commonly written in Dirac notation $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, with $|0\rangle = (1,0)^T$, $|1\rangle = (0,1)^T$, with $\alpha^2 + \beta^2 = 1$. The complex conjugate of $\langle\psi|$ is denoted as $\langle\psi| = |\psi\rangle^\dagger$. The Hilbert space of N qubits is formed by the tensor product $\otimes$ of N single-qubit spaces with dimension $d = 2^N$. We denote the inner product of two states $|\phi\rangle$ and $|\psi\rangle$ as $\langle\phi|\psi\rangle$ and the overlap is defined as $|\langle\phi|\psi\rangle|$. Quantum gates are unitary matrices, which transform quantum states via the matrix-vector multiplication. Common single-qubit rotation gates include $R_x(\theta) = e^{-i\theta X/2}$, $R_y(\theta) = e^{-i\theta Y/2}$, $R_z(\theta) = e^{-i\theta Z/2}$, which are in the matrix exponential form of Pauli matrices

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{1}$$

B. Quantum machine learning

There are four strategies of how to combine machine learning and quantum computing, based on if one considers the data was created by a classical (C) or quantum (Q) system, and whether the computer that processes data are classical (C) or quantum (Q),(as shown in Figure 1). The scenario CC refers to data that is treated traditionally. This is the traditional technique of machine learning. The scenario QC examines how machine learning might aid quantum computing. The scenario CQ uses quantum computing to examine conventional datasets. The last scenario QQ examines the processing of "quantum data" by a quantum device. Among these approaches, CQ scenario is currently focused by machine learning research community [1].

Quantum Machine Learning (QML) algorithms are separated into three distinct strategies. Quantum machine learning

[1]https://www.ibm.com/quantum/qiskit

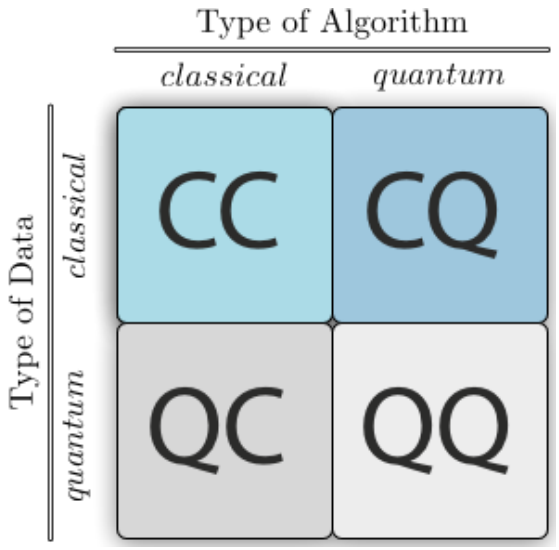

Fig. 1. Four different ways to combine quantum computing with machine learning. Source from [10] [11].

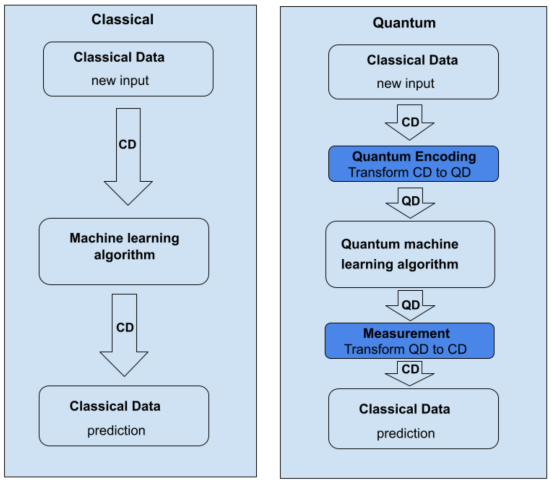

Fig. 2. Processing techniques of conventional machine learning and quantum machine learning. CD represents classical data and QD represents quantum data.

algorithms, which are quantum versions of classical machine learning, in addition to algorithms that are implemented on a real quantum computers such as QSVM, Quantum Neural Network, and Quantum Linear Regression. The second strategy is quantum-inspired machine learning, which leverages the concepts of quantum computing to enhance traditional machine learning algorithms. The third strategy, hybrid classical-quantum machine learning algorithms, merges classical and quantum algorithms to improve performance and reduce the cost of learning—for example, using quantum circuit to propose a novel variational quantum classifier.

Figure 2 illustrates the processing strategies used in conventional machine learning and QML. In traditional machine learning, data are a direct input to the algorithm, which then analyses the data and generates an output. QML, on the other hand, demands the initial encoding of the data into quantum data. QML receives quantum data as input, processes it, and generates quantum data as output. The quantum data are then converted to conventional data.

III. Quantum Neural Network

A. Variational Quantum Algorithms

Variational Quantum Algorithms (VQAs) combine the strengths of quantum circuits with classical optimization techniques. They operate in a hybrid fashion, alternating between a quantum circuit (often a parameterized quantum circuit) and a classical optimizer. The primary objective is to determine optimal parameters that either minimize or maximize a given objective function.

As schematically shown in Figure 3, the first step to developing a VQA is to define a cost (or loss) function C which encodes the solution to the problem. One then proposes an ansatz, that is, a quantum operation depending on a set of continuous or discrete parameters θ that can be optimized. This ansatz is then trained in a hybrid quantum-classical loop to solve the optimization task.

$$\theta^* = \text{argmin}_\theta C(\theta) \tag{2}$$

1) Cost function: A crucial aspect of a VQA is encoding the problem into a cost function. Similar to classical machine learning, the cost function maps values of the trainable parameters θ to real numbers. More abstractly, the cost defines a hyper-surface usually called the cost landscape such that the task of the optimizer is to navigate through the landscape and find the global minima. Without loss of generality, the cost can be expressed as

$$C(\theta) = f(\rho_k, O_k, U(\theta)) \tag{3}$$

where f is some function, $U(\theta)$ is a parametrized unitary, θ is composed of discrete and continuous parameters, ρ_k are input states from a training set, O_k are a set of observables. Often it is useful, and possible, to express the cost in the form

$$C(\theta) = \sum_k f_k(Tr[O_k U(\theta)\rho_k U^\dagger(\theta)]) \tag{4}$$

for some set of functions $\{f_k\}$. During the optimization, one uses a finite statistic estimator of the cost or its gradients.

Desirable criteria that the cost function should meet includes "faithful" and "efficiently estimate". For faithful criteria, the cost must be faithful in that the minimum of $C(\theta)$ corresponds to the solution of the problem. Second, one must be able to "efficiently estimate", $C(\theta)$ by performing measurements on a quantum computer and possibly performing classical post-processing. An implicit assumption here is that the cost should not be efficiently computable with a classical computer, as this would imply that no quantum advantage can be achieved with the VQA. In addition, it is also useful for $C(\theta)$ to be "operationally meaningful", so that smaller cost values indicate a

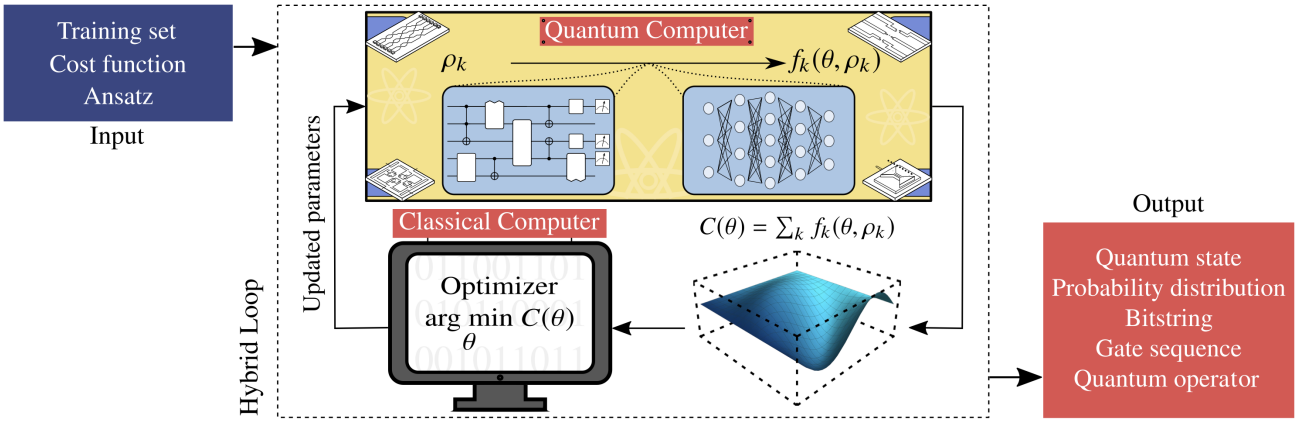

Fig. 3. Schematic diagram of a Variational Quantum Algorithm (VQA) (source from [12])

better solution quality. Finally, the cost must be "trainable", which means that it should be possible to efficiently optimize the parameters θ.

2) Ansatzes: Another important aspect of a VQA is its ansatz. Generically speaking the form of the ansatz dictates what the parameters θ are, and hence, how they can be trained to minimize the cost. The specific structure of an ansatz will generally depend on the task at hand, as in many cases one can use information about the problem to tailor an ansatz. These are the so-called "problem-inspired ansatze". However, some ansatz architectures are generic and "problem-agnostic", meaning that they can be used even when no relevant information is readily available. For the cost function in 4, the parameters can be encoded in a unitary $U(\theta)$ that is applied to the input states to the quantum circuit. $U(\theta)$ can be generically expressed as the product of L sequentially applied unitary operators:

$$U(\theta) = U_L(\theta_L) \dots U_1(\theta_1) \tag{5}$$

with

$$U_L(\theta_L) = \prod_m e^{-i\theta_m H_m W_m} \tag{6}$$

, W_m is an unparametrized unitary and H_m is a Hermitian operator; θ_l is the l-th element in θ.

3) Gradients descent: Once the cost function and ansatz have been defined, the next step is to train the parameters θ and solve the optimization problem of 4. It is known that for many optimization tasks using information in the cost function gradient (or in higher-order derivatives) can help in speeding up and guaranteeing the convergence of the optimizer. One of the main advantages of many VQAs is that, one can analytically evaluate the cost function gradient. The well-known methods is Parameter-shift rules [13].

B. Quantum neural network

Quantum neural networks belong to the family of variational quantum algorithms [14]. They consist of quantum circuits with parameterized gate operations. Initially, information is encoded into a quantum state using a state preparation routine or feature map [15]. The choice of feature map aims to enhance the quantum model's performance, although it is typically not optimized or trained. However, this concept has been discussed in prior research [16]. Once data is encoded, a variational model with parameterized gates is applied and optimized for a specific task [17]. This optimization involves minimizing a loss function, and the quantum model's output can be extracted using a classical post-processing function applied to measurement outcomes.

QNNs can be viewed from two perspectives:

- **Machine Learning Perspective**: Quantum Neural Networks (QNNs) are algorithmic models that discover hidden patterns in data, similar to classical counterparts. They process classical inputs using quantum gates with trainable weights, and the resulting state is used to train the weights via backpropagation.
- **Quantum Computing Perspective**: QNNs are quantum algorithms based on parametrized quantum circuits. These circuits include a feature map (with input parameters) and an ansatz (with trainable weights), and they can be trained variationally using classical optimizers.

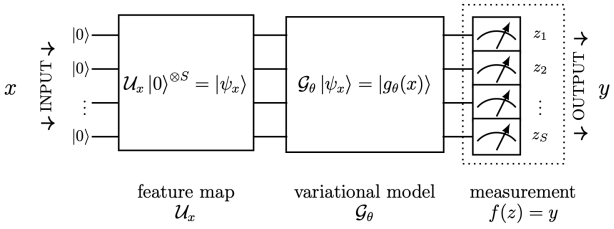

Fig. 4. General structure of quantum neural network (source [18]). the input $x \in \mathbb{R}^{S_{\text{in}}}$ is encoded into an S-qubit Hilbert space by applying the feature map $|\psi\rangle_x := \mathcal{U}_x |0\rangle^{\otimes S}$. This state is then evolved via a variational form, where the parameters $\theta \in \Theta$ are chosen to minmize a certain loss function. Finally, a measurement is performed whose outcome $z = (z_1, \dots, z_S)$ is post-processed to extract the output of the model $y := f(z)$.

The Quantum Neural Network (QNN) model, illustrated in Figure 4, transforms classical data represented by $x \in \mathbb{R}^{S_{\text{in}}}$ into an S-qubit Hilbert space using the feature map $\mathcal{U}_x$ [19]

. Initially, Hadamard gates act on each qubit. Subsequently, normalized feature values are encoded using RZ-gates, with rotation angles directly tied to the data's feature values. Additionally, RZZ-gates capture higher-order interactions, where the controlled rotation angles depend on the product of feature values. These encoding steps can be iterated an arbitrary number of times, with the depth of the feature map determining the extent of repetition. Once the data is encoded, the model optimizes a variational circuit $G\theta$ comprising parameterized RY-gates and CNOT entangling layers between qubit pairs. The trainable parameters θ are adjusted during optimization. Finally, a post-processing step measures all qubits in the σ_z basis and computes the parity of the resulting bit strings.

For instance, it has been shown that quantum neural networks can achieve a significantly higher capacity, as measured by the effective dimension, than comparable classical neural networks [18], implying that the former can express a broader class of functions than the latter. Moreover, it has also been pointed out that quantum algorithms can outperform classical ones in deep learning problems [20], potentially provide exponentially better ability to generalize when trained to predict the outcome of physical processes [21], and more recently a VQA has been proposed for deep reinforcement learning [22]. An exciting prospect for using quantum neural networks is that certain architectures are immune to barren plateaus, and hence are trainable even for large problems [23], [24].

IV. Numerical studies

A. Benchmarks

TABLE I
STATISTICS ON DATA USING IN THE EXPERIMENTS. COLUMNS C, D, N, N_{TRAIN}, AND N_{TEST} DENOTES FOR NUMBER CLASSES, NUMBER OF FEATURES, TOTAL NUMBER OF SAMPLES, NUMBER OF TRAINING SAMPLES, AND NUMBER OF TESTING SAMPLES.

Dataset	C	N	N_{train}	N_{test}
MNIST-2	2	100	90	10
FashionMNIST-2	2	100	90	10
FashionMNIST-3	3	150	135	15
Syn-Dataset-4	2	100	90	10

Two common classification datasets: MNIST [25] and Fashion MNIST [26] were chosen as benchmark to validate performance of models. Especially, for MNIST, two classes (i.e., digits 3 and 6) would be extracted, for Fashion-MNIST, classes dress, shirt, T-shirt/top would be used and denoted as "Fashion-MNIST-2", "Fashion-MNIST-3". The information for all dataset were listed in the table I.

Besides, synthesis datasets would be used in the project denoted as "Syn-Dataset-4" to indicate data with 4 features. This dataset contains two classes, for the class "0" we generate first two features follow the white noise distribution $\mathcal{N}(0, 1)$, and the last two features follow the $\mathcal{N}(0, 2)$; while for the class "1", we generate on the opposite way.

To summarize, all datasets contains samples of 4 dimensions, and randomly select 50 samples/class as the entire. The splitting ratio between training and testing dataset is 90%, and 10%.

B. Models

To make a comparison, we evaluate the performance of traditional machine learning models: Naive Bayes classifier, Support Vector Machinesl; and (classically) deep learning learning model, which is stacked by fully connected layers. To answer how the size of model , or the depth of network effect on the performance, we design two models for DNN models which are called simpleANN and complexANN. For the QNNs, we use the ZZFeatureMap[2] as quantum encoding, and EfficientSU2[3] as hardware ansatz. Similar to the DNNs models, we design two QNN models: simple QNN which contains only one ansatz, and complex QNN which contains three replications of the ansatz. The statistics of testing models were shown in table II.

TABLE II
STATISTICS ON MODELS USING IN THE STUDY. IN THAT, C, AND D DENOTE FOR THE NUMBER OF CLASSES IN THE DATASET, AND FEATURE DIMENSION OF INPUT SAMPLES RESPECTIVELY.

Model	# trainable params
NBC	$2 \times C \times D$
SVM	$2 \times (C-1) \times D$
SimpleANN	$(D+1) \times C$
ComplexANN	$20768 + (D+1) \times C$
SimpleQNN	$4 \times D$
CompleQNN	$8 \times D$

For training classical deep learning models, we fixed the number of epochs is 30, and for training VQC models, we let it run 50 epochs. For the hardware usage, we utilize the Google Colab GPU Tesla T4[4] card contains 40 SMs with a 6MB L2 cache shared by all SMs. It also ships with 16GB high-bandwidth memory (GDDR6) that is connected to the processor.

C. Evaluation metrics

To measure the effectiveness of models, we use the accuracy reported on testing dataset. During the training stage, we do not let models use the testing dataset to update parameters. Since the balance between the number of samples in each class, we only consider the accuracy metric:

$$\text{acc} = \frac{\#\text{corrected predictions}}{\#\text{predictions}} \tag{7}$$

D. Results

Figure 5 shows the performances of models on the entire training datasets. In almost cases, the ComplexANN is the best one, except on the "FashionMNIST-3", showing its flexibility to fit on different datasets from simple to complex. On the "MNIST-2", the classical machine learning group gain performance approximately to the ComplexANN while simpleANN performs extremely poor which suggess us that the simpleANN is not complex enough to capture the structure of data. The interesting point here is the performance of

[2]https://docs.quantum.ibm.com/api/qiskit/qiskit.circuit.library.ZZFeatureMap
[3]https://docs.quantum.ibm.com/api/qiskit/qiskit.circuit.library.EfficientSU2
[4]

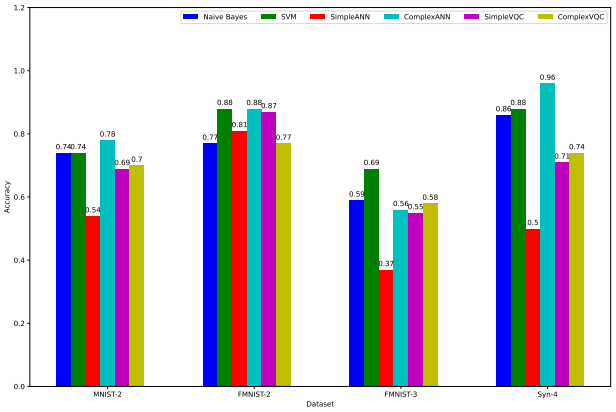

Fig. 5. Training accuracy of models.

VQC models are quite comparable with the performance of ComplexANN, especially on the "Fashion-MNIST-2", and "Fashion-MNIST-3". However, on "Fashion-MNIST-3", NB, and SVM perform better than other ones showing us that these deep learning models, and VQCs is not complex enough. Also, the training accuracy of VQCs model on the "Syn-Dataset-4" is smaller than the classical machine learning group which give us the idea to increase the depth of deep network.

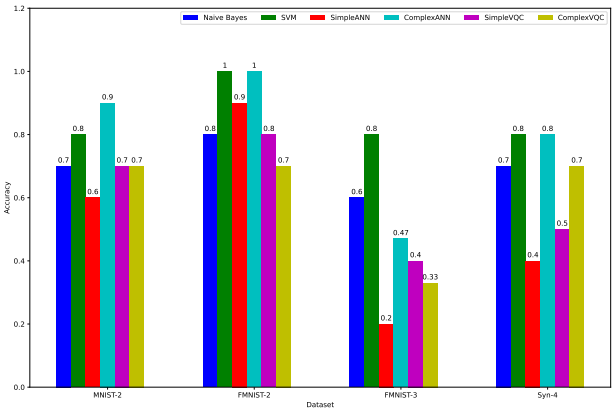

Fig. 6. Testing accuracy of models.

After observing the training performance of these models, we evaluated these models on on testing subsets, which were illustrated via figure 6. From the graphs, we can see that the same schema on the training performance. In particular, the performance of ComplexANN is the best one in almost cases, except the FashionMNIST-3. It shows us that the complexity of this dataset is non-trivial. In that, the performance of ComplexVQX on testing datasets are $0.7, 0.7, 0.33, 0.7$ (from left to right). Also, the performance of SimpleVQC is better than performance of SimpleANN in all cases (also on training subsets) shows us potential advantages of QNNs which is achieving a better, or equal performance when comparing with the deep learning models having the same complexity.

V. Conclusion

In this project, we explored the quantum neural network model, which is a raising field in recent year. The advantages of this subject lie in the intersection between doing machine learning on quantum hardware. We presented the theory, components of Variational quantum algorithm which is the root of quantum neural network. Then, we experimented the traditional machine learning models including Naive Bayes, SVMs, DNNs, and VQCs (a type of QNNs) on various benchmarks. From the numerical results, the practical performance of VQCs are still slower than traditional machine learning models, and deep learning models. However, the it is quite a good approximation to these performance showing the further investigation for improving the performance of VQCs on real-world datasets.

References

[1] A. Zeguendry, Z. Jarir, and M. Quafafou, "Quantum machine learning: A review and case studies," *Entropy*, vol. 25, no. 2, p. 287, 2023.

[2] D. Peral-García, J. Cruz-Benito, and F. J. García-Peñalvo, "Systematic literature review: Quantum machine learning and its applications," *Computer Science Review*, vol. 51, p. 100619, 2024.

[3] M.-G. Zhou, Z.-P. Liu, H.-L. Yin, C.-L. Li, T.-K. Xu, and Z.-B. Chen, "Quantum neural network for quantum neural computing," *Research*, vol. 6, p. 0134, 2023.

[4] J. Tian, X. Sun, Y. Du, S. Zhao, Q. Liu, K. Zhang, W. Yi, W. Huang, C. Wang, X. Wu *et al.*, "Recent advances for quantum neural networks in generative learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[5] R. Correll, S. J. Weinberg, F. Sanches, T. Ide, and T. Suzuki, "Quantum neural networks for a supply chain logistics application," *Advanced Quantum Technologies*, vol. 6, no. 7, p. 2200183, 2023.

[6] L.-H. Gong, J.-J. Pei, T.-F. Zhang, and N.-R. Zhou, "Quantum convolutional neural network based on variational quantum circuits," *Optics Communications*, vol. 550, p. 129993, 2024.

[7] P. Adebayo, F. Basaky, and E. Osaghae, "Variational quantum-classical algorithms: A review of theory, applications, and opportunities," *UMYU Scientifica*, vol. 2, no. 4, pp. 65–75, 2023.

[8] Z. Hu, P. Dong, Z. Wang, Y. Lin, Y. Wang, and W. Jiang, "Quantum neural network compression," in *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, 2022, pp. 1–9.

[9] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. Cambridge university press, 2010.

[10] V. Dunjko, J. M. Taylor, and H. J. Briegel, "Quantum-enhanced machine learning," *Physical review letters*, vol. 117, no. 13, p. 130501, 2016.

[11] E. Aïmeur, G. Brassard, and S. Gambs, "Machine learning in a quantum world," in *Advances in Artificial Intelligence: 19th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2006, Québec City, Québec, Canada, June 7-9, 2006. Proceedings 19*. Springer, 2006, pp. 431–442.

[12] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.

[13] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," *Physical Review A*, vol. 99, no. 3, p. 032331, 2019.

[14] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, "Circuit-centric quantum classifiers," *Physical Review A*, vol. 101, no. 3, p. 032308, 2020.

[15] M. Schuld, R. Sweke, and J. J. Meyer, "Effect of data encoding on the expressive power of variational quantum-machine-learning models," *Physical Review A*, vol. 103, no. 3, p. 032430, 2021.

[16] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, "Quantum embeddings for machine learning," *arXiv preprint arXiv:2001.03622*, 2020.

[17] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Physics*, vol. 15, no. 12, pp. 1273–1278, 2019.

[18] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, "The power of quantum neural networks," *Nature Computational Science*, vol. 1, no. 6, pp. 403–409, 2021.
[19] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, 2019.
[20] N. Wiebe, A. Kapoor, and K. M. Svore, "Quantum deep learning," *arXiv preprint arXiv:1412.3489*, 2014.
[21] H.-Y. Huang, R. Kueng, and J. Preskill, "Information-theoretic bounds on quantum advantage in machine learning," *Physical Review Letters*, vol. 126, no. 19, p. 190505, 2021.
[22] S. Y.-C. Chen, C.-H. H. Yang, J. Qi, P.-Y. Chen, X. Ma, and H.-S. Goan, "Variational quantum circuits for deep reinforcement learning," *IEEE access*, vol. 8, pp. 141 007–141 024, 2020.
[23] H.-k. Zhang, C. Zhu, M. Jing, and X. Wang, "Statistical analysis of quantum state learning process in quantum neural networks," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
[24] A. Pesah, M. Cerezo, S. Wang, T. Volkoff, A. T. Sornborger, and P. J. Coles, "Absence of barren plateaus in quantum convolutional neural networks," *Physical Review X*, vol. 11, no. 4, p. 041011, 2021.
[25] A. Baldominos, Y. Saez, and P. Isasi, "A survey of handwritten character recognition with mnist and emnist," *Applied Sciences*, vol. 9, no. 15, p. 3169, 2019.
[26] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.
[27] Y. He and L. Xiao, "Structured pruning for deep convolutional neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
[28] B. Rokh, A. Azarpeyvand, and A. Khanteymoori, "A comprehensive survey on model quantization for deep neural networks in image classification," *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 6, pp. 1–50, 2023.
[29] M. Yuan, J. Bai, F. Jiang, and L. Du, "A systematic dnn weight pruning framework based on symmetric accelerated stochastic admm," *Neurocomputing*, vol. 575, p. 127327, 2024.

APPENDIX A
CODE AND DATA

All the datasets and source code were presented in this work can be found at https://drive.google.com/drive/folders/1_sk2Cz3qiZxO_SNWcnKJ5D6pg5opzb6T?usp=share_link

APPENDIX B
DEEP NEURAL NETWORK COMPRESSION

Pruning [27] and quantization [28] are effective in reducing the model size and speeding up its execution, which has been thoroughly explored for classical DNNs. With the powerful ADMM [29] optimization framework, many researchers achieve a very high compression ratio while maintaining high performance.

APPENDIX C
QUANTUM NEURAL NETWORK COMPRESSION

Variational Quantum Circuit [6] is one of the type of QNNs that is known through existing works that have theoretically shown the effectiveness and efficiency of VQC in the perfect scenario [7]. However, it is well-known that near-term quantum devices have high noise. Coherent and incoherent errors in quantum devices will in turn limit the quantum circuit depth/length (i.e., a longer circuit has larger accumulated errors) As such, when it comes to deploying QNN onto the near-term quantum devices, it is critical to control the circuit depth which poses a significant challenge. It not only increases the computational complexity but also introduces more opportunities for errors due to quantum decoherence.

To tackle the problem above, [8] proposed the CompVQC framework, which mimics and also incorporates the pruning, and quantization techniques which we already know in classical deep neural networks, to reduce the circuit depth, but also maintains a good performance.

A. Quantum Neural Network Compression framework

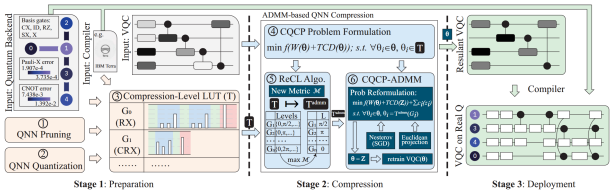

Fig. 7. Overview of CompVQC [8] to compress and deploy quantum neural networks via 3 stages.

Figure 7 shows the pipeline of the Variational Quantum Circuit Compression framework (CompVQC). Given a quantum device and a quantum circuit of QNN, CompVQC [8] passes through 3 stages to compress a QNN and deploys the compressed QNN to the quantum device.

First, the preparation stage is run to provides several prior knowledge to support the compression of QNN models. The outcome of this stage is the compression deploys-level LUT, which will be the base to support the compression in stage 2.

- QNN pruning: device ignore certain quantum gates based on the weight θ, and multiple factors such as identify operators.
- QNN quantization: as the technique using low-bit variables in classical computing, we can expect to reduce the circuit depth of QNNs similarly. However, parameters in quantum computing indicate the rotation of a quantum state, and the data type will not affect the circuit depth.
- Compression-Level look up-table: Different quantum gates have varied pruning and quantization levels.

Then the 2nd stage, the compression progress will actually run by sequentially oparate the following optimization problems: Compilation-aware QNN Compression Problem (CQCP), Compression-Level LUT Reconstruction for ADMM, and ADMM-Based Compilation-aware QNN Compression. This stage utilize mainly the ADMM algorithm to introducing removable parameters, which is the main block in CompVQC framework.

At the end, for the deployment, or inference (3rd stage), CompVQC will leverage the input compiler to do the translation that maps the logical quantum circuit to the given quantum device.